Abstract—Video summarization is a simplification of video content for compacting the video information. The video summarization problem can be transformed to a clustering problem, in which some frames are selected to saliently represent the video content. In this work, we use a hierarchical graph-based clustering method for computing a video summary. In fact, the proposed approach, called HSummary, adopts a hierarchical clustering method to generate a weight map from the frame similarity graph in which the clusters (or connected components of the graph) can easily be inferred. Moreover, the use of this strategy allows to apply a similarity measure between clusters during graph partition, instead of considering only the similarity between isolated frames. Furthermore, a new evaluation measure that assesses the diversity of opinions of user summaries, called Covering, is also proposed. Experimental results provide quantitative and qualitative comparison between the new approach and other popular algorithms from the literature, showing that the new algorithm is robust and efficient. Concerning quality measures, HSummary outperforms the compared methods regardless of the visual feature used in terms of F-measure.

Keywords—Graph-based hierarchical video summarization; covering; global descriptors; observation scales.

I. INTRODUCTION

The increasing number of video files has made the task of searching for a specific content very expensive, because it is necessary to index the video information. Usually, there are two approaches to cope with the index problem: (i) manual annotation; or (ii) automatic annotation. The former is expensive and subjective, since it depends on the experts to perform this annotation. The second one is objective and is directly related to the visual contents, however it depends on the features which are used. The cost to find a specific content related to a video depends on the size of the index, thus instead of considering all video content, one can summarize it in order to reduce the search space. In literature, there are many approaches to simplify the video content [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Thus, video summarization is a simplification of video content in order to reduce the amount of data without losing information. The video summarization problem can be transformed into a clustering problem, in which some frames are selected to saliently represent the video content, as illustrated in Fig. 1.

In [5], the authors proposed an approach to cope with the video summarization problem in which the clustering is achieved by a K-means algorithm, but it is necessary to know a priori the number of clusters. In [8], it was proposed the use of graph-theoretic FCM algorithm for video summarization, however the graph creation is directly related to number of centers. In [6], the authors used a Delaunay triangulation to automatically identify the frame clusters, however this approach is expensive and produces very compressed summaries. VISTO [7] is based on low-level video frames color feature extraction and on a modification of furtherest point-first algorithm to cluster the frames. This approach is fast but the summaries are big. In [9], the authors use a graph-theoretic divisive clustering algorithm based on construction of a Minimum Spanning Tree (MST) to select video frames without segmenting the video into shots or scenes, eliminating pre-processing steps. It is important to note that according to [12] the MST approach for clustering is hierarchical, and thanks to this property, it is easy to compute a video summary regarding the specified number of keyframes. Recently, some new approaches for video summarization using constraint satisfaction programming [10] and prior information [11] were proposed.

In this work, video summarization problem is also transformed into a graph-based clustering problem in which a cluster (or connected component of the graph), computed from a graph partition, represents a set of similar frames. Similar to [9], the proposed method – HSummary – elimi-
nates the pre-processing steps using a MST. But instead of computing the graph partition directly over the MST, the proposed approach adopts a hierarchical clustering method to generate a weight map from the MST in which the clusters can easily be inferred. Moreover, the use of a hierarchical graph-based clustering method allows HSummary to apply a similarity measure between clusters (i.e., set of frames) during graph partition, instead of considering only the similarity between isolated frames. Finally, a new evaluation measure, called Covering, is also proposed. This measure assesses the diversity of opinions of user summaries, what is not take into account when average values of quality measures are used [5]. The paper is organized as follows. Section II describes the clustering problem using minimum spanning tree and also defines many concepts used in our work. Section III describes our methodology to solve the video summarization problem, while Section IV presents a new evaluation measure that can be used to compare video summarization methods. Section V describes the performed experiments together with a comparative analysis between our approach and the others methods. Finally, we give some conclusions in Section VI.

II. SOME FUNDAMENTAL CONCEPTS

A. Similarity graph

Let \( \mathbb{A} \subset \mathbb{N}^2 \), \( \mathbb{A} = \{0, \ldots, H - 1\} \times \{0, \ldots, W - 1\} \), where \( H \) and \( W \) are the width and height of each frame, respectively, and, \( \mathbb{T} \subset \mathbb{N} \), \( \mathbb{T} = \{0, \ldots, N - 1\} \), in which \( N \) is number of frames of a video. A frame \( f \) is a function from \( \mathbb{A} \) to \( \mathbb{R}^3 \), where for each spatial position \((x,y)\) in \( \mathbb{A} \), \( f(x,y) \) represents the color value at pixel location \((x,y)\).

A video \( V_N \), in domain \( \mathbb{A} \times \mathbb{T} \), can be seen as a sequence of frames \( f \). It can be described by \( V_N = \{f(t)\}_{t \in \mathbb{T}} \), where \( N \) is the number of frames contained in the video. A frame is usually described in terms of global descriptors, such as color histogram or GIST [13].

**Definition 1 (Frame similarity)** Let \( f_{t_1} \) and \( f_{t_2} \) be two video frames at locations \( t_1 \) and \( t_2 \), respectively. Two frames are similar if a distance measure \( D(f_{t_1}, f_{t_2}) \) between them is smaller than a specified threshold (\( \delta \)). The frame similarity is defined as

\[
FS(f_{t_1}, f_{t_2}, \delta) = \begin{cases} 
1, & \text{if } D(f_{t_1}, f_{t_2}) \leq \delta \\
0, & \text{otherwise}
\end{cases}
\]  

(1)

There are several choices for global measures \( D(f_{t_1}, f_{t_2}) \), i.e., the distance measure between two frames, e.g., histogram/frame difference, histogram intersection, difference of histograms means, and others. For the global descriptor GIST, the representations are usually compared using \( L_2 \) norm. After selecting one, it is possible to construct a frame similarity graph based on a video \( V_N \) and a distance measure as follows.

**Definition 2 (Frame similarity graph – \( G^\delta \))** Let \( V_N \) be a video with \( N \) frames. A frame similarity graph \( G^\delta = (V^\delta, E^\delta) \) is a weighted undirected graph. Each node \( v_{t_1} \in V^\delta \) represents a frame \( f_{t_1} \in V_N \). There is an edge \( e \in E^\delta \) with weight \( w(e) = D(f_{t_1}, f_{t_2}) \) between two nodes \( v_{t_1} \) and \( v_{t_2} \), if frame similarity of associated frames is equal to \( 1 \):

\[
E^\delta = \{ (v_{t_1}, v_{t_2}, D(f_{t_1}, f_{t_2})) \mid v_{t_1} \in N, v_{t_2} \in V^\delta, \quad FS(f_{t_1}, f_{t_2}, \delta) = 1 \}
\]  

(2)

Fig. 5(a) illustrates a frame similarity graph of a real video in which only the frames 1, 501, 1001, 1501, 2001, 2501 and 3001 are sampled. The similarity measure used is the histogram intersection in HSV color space normalized in the range \([0, 100]\).

**Definition 3 (Tree of frames – \( fT_{G^\delta} \))** Let \( G^\delta = (V^\delta, E^\delta) \) be a frame similarity graph. An edge-weighted tree of frames \( fT_{G^\delta} = (V^\delta, E^\delta_2) \) is a connected acyclic subgraph of \( G^\delta \), i.e., \( E^\delta_2 \subseteq E^\delta \). The weight of \( fT_{G^\delta} \) is equal to the sum of weights of all edges belonging to \( E^\delta_2 \), i.e., \( w(fT_{G^\delta}) = \sum_{e \in E^\delta_2} w(e) \).

**Definition 4 (Minimum spanning tree of frames – \( fMST_{G^\delta} \))** Let \( G^\delta = (V^\delta, E^\delta) \) be a frame similarity graph. The minimum spanning tree of frames \( fMST_{G^\delta} \) is a tree of frames whose weight is minimal.
According to [15], a $k$-clustering divides the elements into $k$ non-empty groups, in which the insertion of an element into a group depends on distance measure between this element and the elements already in the group. In order to compute the video summarization, a $k$-clustering divides the video sequence into $k$ video scenes, and, consequently, it is necessary to eliminate $k - 1$ edges from the MST.

The edge removal operation, when applied to a tree, produces two connected components. Here, each connected component is called frame cluster, as defined to follow. The process of edge removal must be agreed to a specified criterion. In [9], it was proposed three different one: (i) removal of largest weight edges; (ii) removal of edges with weight greater than or equal to a specified threshold; or (iii) removal of the largest weight edge if this edge does not verify some properties. The former is useful when the number of clusters (defined before) is pre-determined. The second one can be considered when the minimum similarity measure between clusters is specified, and can be considered a special case of the first one when the weights are different. The latter is useful for producing homogeneous components (or clusters). For example, the removal in Fig. 5 of all edges with weight greater than or equal to 49 produces the same result when we eliminate the 2 largest weight edges.

**Definition 5 (Set of frame clusters – $C^{*,k}$)** Let $fT_{G^δ}$ be an edge-weighted tree of frames. Let $C^{*,k}$ denote the set of $k$ connected components $C^{*,k}_1, C^{*,k}_2, \ldots, C^{*,k}_k$ formed by deleting the $k - 1$ largest edges of $fT_{G^δ}$ in which $C^{*,k}_i = (V^{δ}_i, E^{δ}_i)$.

There are different ways for computing the tree of frames $fT_{G^δ}$ which is used in generating a set of frame clusters $C^{*,k}$, e.g., the minimum spanning tree of frames $fMST_{G^δ}$ was used in [9]. But, in this work, a weight map – an edge-weighted tree of hierarchical observation scales – is adopted during the calculation of the set of frame clusters. The construction of this weight map follows the hierarchical approach describe in [16] for partitioning a graph using observation scales. More details about the generation of this scale map could be found in Subsection II-B.

The number of clusters, and consequently, the number of video scenes is directly related to the number of edge removal operations when a tree is considered. Also, the process to compute the frame cluster is hierarchical in the sense that the edge removal divides a cluster into two different groups. It is important to note that two clusters will never be merged after application of edge removal operation. However, the saliency of a frame cluster component may depend on its size, since components with a small number of frames may represent noise. For example, a black frame or a flashlight frame are probably very dissimilar of all other frames and, consequently, the adjacent edges will be the largest weight edges of the $fT_{G^δ}$. The frame cluster produced by edge removal may present a very small number of frames, thus it can be ignored in subsequent analysis. In order to illustrate the content of a frame cluster, the video frame which is more similar to the other frames belonging to same the cluster must be chosen. For that, the center of a frame cluster is defined as follows.

**Definition 6 (Center of the frame cluster – $S^{*,k}_i$)** Let $C^{*,k}_i$ be a frame cluster. The center $S^{*,k}_i$ of the frame cluster $C^{*,k}_i$ is the frame which is more similar to the other frames belonging to same the cluster $C^{*,k}_i$.

Usually, the center of the frame cluster may be computed by two different ways: (i) unweighted center selection; or (ii) weighted center selection. In the former, the center is associated to the frame which is the most central, in terms of number of frames to the other ones. In the latter, the most central frame in terms of dissimilarity measure distances is selected.

**B. A hierarchical graph-based clustering method**

Thanks to the method proposed in [16], one can compute the hierarchical observation scales using the method called cp-HOScale (or simply HOScale), in which the adjacent regions that are evaluated depend on the order of the merging in the fusion tree (or simply the order of the merging between connected components on the minimum spanning tree – $fMST_{G^δ}$ – of the original graph). Generally speaking, a new edge-weighted tree is created from this $fMST_{G^δ}$ in which each edge weight corresponds to the scale from which two adjacent regions connected by this edge are correctly merged, i.e., there are no two sub-regions of these regions that might be merged before these regions. For computing the new weight map, we consider the criterion for region-merging proposed in [17] which measures the evidence for a boundary between two regions by comparing two quantities: one based on intensity differences across the boundary, and the other based on intensity differences between neighboring pixels within each region. More precisely, in order to know whether two regions must be merged, two measures are considered. The **internal difference** $Int(X)$ of a region $X$ is the highest edge weight among all the edges linking two vertices of $X$ in the $fMST_{G^δ}$. The **difference** $Diff(X,Y)$ between two neighboring regions $X$ and $Y$ is the smallest edge weight among all the edges that link $X$ to $Y$. Then, two regions $X$ and $Y$ are merged when:

$$Diff(X,Y) \leq \min\{Int(X) + \frac{\lambda}{|X|}, Int(Y) + \frac{\lambda}{|Y|}\} \quad (3)$$

where $\lambda$ is a parameter used to prevent the merging of large regions (i.e., larger $\lambda$ forces smaller regions to be merged).

The merging criterion defined by Eq. (3) depends on the scale $\lambda$ at which the regions $X$ and $Y$ are observed. More precisely, let us consider the (observation) scale $S_Y(X)$ of $X$ relative to $Y$ as a measure based on the difference between $X$ and $Y$, on the internal difference of $X$ and on
the size of $X$:

$$S_Y(X) = (\text{Diff}(X, Y) - \text{Int}(X)) \times |X|.$$  \hfill (4)

Then, the scale $S(X, Y)$ is simply defined as:

$$S(X, Y) = \max(S_Y(X), S_X(Y)).$$ \hfill (5)

Thanks to this notion of a scale, Eq. (3) can be written as:

$$\lambda \geq S(X, Y).$$ \hfill (6)

The core of HOScale [16] is the identification of the smaller scale value that can be used to merge the largest region to another region while guaranteeing that the internal differences of these merged regions are larger than the value calculated for the smaller scale. In fact, instead of computing the hierarchy of partitions, a weight map constructed using the notion of scale presented in Eq. (3) is produced from which the desired hierarchy can be inferred, e.g., by removing from those edges whose weight is greater than the desired scale. In other words, frame clusters are easily identified after edge removal.

III. GRAPH-BASED HIERARCHICAL VIDEO SUMMARIZATION

Usually, video summarization methods are based on clustering algorithms, as in [5], [9]. Even though these methods have provided very interesting results, there are still some issues to be better explored such as: (i) the recommended size for a video summary; and (ii) the properties of the generated video summary. Another very important issue is related to the evaluation criterion be used to compare the different summarization methods (this it will be discussed in details in Sec. IV).

Both issues listed above are directly related to the method that is used to partition the video, and, consequently, to produce the video summary. In [16], it was proposed a hierarchical method for partitioning a graph using observation scales. As expected, the result of this method is a hierarchy of partitions in which the clusters can easily be inferred. In this work, video summarization problem is transformed into a graph-based clustering problem in which a cluster (or connected component of the graph), computed from a graph partition, represents a set of similar frames. Fig 3 illustrates the proposed method for video summarization, so-called HSummary. The method depends basically on: (i) the visual feature selected for representing a frame; (ii) the dissimilarity measure used for comparing two frames; (iii) the minimum size of a connected component for eliminating outliers; and (iv) the number of connected components for defining the video summary size.

Thanks to adoption of the minimum spanning tree, the proposed method eliminates the pre-processing to compute the number of clusters, and also, eliminates the video segmentation step. Thus, the minimum spanning tree of frames is computed from the frame similarity graph that was generated from the video sequence. Afterwards, the observation scales are calculated in order to allow the use of a hierarchical clustering method. It is important to note that the clusters, represented by connected components on the tree, can be obtained by deleting of edges from the weight map until stability. The concept of stability is related to two possibilities: (i) the number of desired clusters; and (ii) the frame similarity inside a cluster. While the former is related to compress factor of the video summary, the second
Comparing User quality of a summary is highly subjective. In this way, some task, since there is no a well-known groundtruth and the in order to assess and compare video summaries. In [7], the manually by a number of users from the sampled frames, Summaries (CUS) centers). cluster is divided into two new clusters (with two new has a new center, but the existence of another change in the set of centers will depend on: (i) the criterion used for centers selection; and (ii) the homogeneity of cluster content. Finally, in Fig. 4d, the clusters obtained into two new clusters but only the recently created cluster has a new center, i.e., the third cluster maintains the same keyframe as center since it is still a good representative of the cluster content. Finally, in Fig. 4d, the clusters obtained by HSummary are shown – the first, the second and the fourth clusters of Fig. 4c remain unchanged, but the third cluster is divided into two new clusters (with two new centers).

IV. A NEW EVALUATION MEASURE

The process of evaluating video summaries is not a simple task, since there is no a well-known groundtruth and the quality of a summary is highly subjective. In this way, some initiatives have been proposed, like Comparison of User Summaries (CUS) [5], in which the video summary is built manually by a number of users from the sampled frames, in order to assess and compare video summaries. In [7], the summaries are compared to Open Video1 summaries (called OV) and are classified into 4 classes: (i) same keyframes; (ii) fewer keyframes; (iii) more keyframes; and (iv) mismatches keyframes. However this classification depends on an single groundtruth video summary, and it may be not a good measure due to the subjectivity of video summaries.

To solve these problems, it is necessary to consider that the groundtruth is composed by various users. The most common measure to evaluate visuals system is the average of precision and recall computed for each user, however they present some distortions due to the consideration of the average function. In [5], the automatic summaries are compared to user summaries to compute two measures: (i) accuracy rate – CUSa; and (ii) error rate – CUSe. These measures can be defined as follows: 

\[ \text{CUSa} = \frac{n_{m_{AS}}}{n_{US}} \]

and 

\[ \text{CUSe} = \frac{n_{m_{AS}}}{n_{US}} \]

in which \( n_{m_{AS}} \) is the number of matching keyframes from automatic summary (AS), \( \pi_{AS} \) is the number of non-matching keyframes from AS and \( n_{US} \) is the number of keyframes from user summary (US). Moreover, the user summaries are the reference summaries and can be considered as the groundtruth. According to [5], the goals of this approach are: (1) to reduce the subjectivity of the evaluation task; (2) to quantify the summary quality and; (3) to allow comparisons among different techniques to be done quickly.

Despite the fact that these measures are interesting, they do not evaluate well the diversity of user summaries. Moreover, due to the average of the measures computed for each user, some distortions may be produced. Actually, CUSa is very sensitive to outliers and does not well evaluate the diversity of user opinions. For example, let two users A and B with the following summaries \( S_A = \{X,Y\} \) and \( S_B = \{M,N,O,P,Q,R,S,T,U,V\} \). Suppose now that three methods compute the following summaries \( AS_1 = \{X,Y\} \), \( AS_2 = \{M,N,O,P,Q,R,S,T,U,V\} \) and \( AS_3 = \{X,M,N,O,P,Q\} \). The CUSa values for these summaries are equal to 0.5. Thus, three completely different summaries present the same values of CUSa.

In order to cope with this kind of problem, we propose a new measure, called Covering – COV, which is based on histogram of user frames to evaluate the covering of user summary by an automatic summary. This measure assesses the diversity of opinions without ignoring the agreement of opinions among users. In other words, CUSa assesses the average value of ratio calculated for each user between the automatic summary and the summary made by that user. While COV assesses the ratio of automatic summary which are similar to all user summaries. Thus, Covering can be defined as follows.

**Definition 7 (Covering)** Let AS be the automatic summary computed by a summarization method. Let US be a set of \( n \) user summaries. The covering of user summaries by an

1http://www.open-video.org
Figure 6. A comparison between several video summarization methods in terms of CUSa, CUSe and COV. For each method, the best F-measure is shown. The ideal method must have CUSa (or COV) equals to 1, and CUSe equals to zero.

The automatic summary is defined as

$$\text{COV} = \frac{\sum_{U \in \mathcal{U}} |M(AS, U)|}{\sum_{U \in \mathcal{U}} |U|}$$

in which, $M(A, B)$ and $|.|$ are the maximum matching between two set of elements A and B, and the cardinality of a set.

For example, the COV values for three automatic summaries presented before – $AS_1$, $AS_2$ and $AS_3$ – are $2/12$, $10/12$ and $6/12$, respectively.

V. Experiments

In order to compare HSummary with other methods, it was used the same dataset adopted in [7] composed by 50 videos in different genres (documentary, lectures, ephemeral, historical, educational). With respect to the assessment, we consider the same measures proposed by [5] in terms of

Comparison of User Summaries (CUS), in which the video summary is built manually by a number of users from the sampled frames. In fact, the summaries are evaluated using CUSa and CUSe.

Unfortunately, using only these two measures, it is not possible to decide which approach is the best method for computing a video summary. In order to solve this issue, a trade-off between the well-known measures in information retrieval, precision and recall, is calculated in terms of harmonic average, called F-measure. The precision is the fraction of detections that are true positives rather than false positives, while recall is the fraction of true positives that are detected rather than missed. Furthermore, we also assess the methods using the new evaluation metric, called Covering, which measures the diversity of opinions. In this work, HSummary is compared to VSUMM1 [5], VSUMM2 [5], AGM1 [9], AGM2 [9], DT [6], VISTO [7] and Open Video summaries (called OV). Note that the parameters are tuned according to original works.

As described in Section III, HSummary depends on: (i) visual feature; (ii) dissimilarity measure; (iii) minimum size of each cluster; and (iv) number of clusters. Concerning visual feature, the behavior of HSummary with two global features is analyzed: (i) using GIST [13] – HSummary$_G$ and; (ii) using BossaNova [18] with SIFT – HSummary$_{BN}$. Note that a mid-level representation, like BossaNova, can be considered as a global descriptor. The $L_2$ norm is used for computing the distance between two frames; and the minimum size of each cluster is set to 10 frames. Finally, for computing video summaries using HSummary, its hierarchical properties are take into account in order to easily compute summaries with different sizes for the same video. Thus, for each video, video summaries with different sizes are assessed in order to maximize the F-measure. This kind of strategy is well-known and adopted in hierarchical image

![Figure 5](image-url)
segmentation for identifying the best parameters for each image to be segmented.

Fig. 6 presents the quantitative measures computed for several methods. For the matching between an automatic summary and an user-made one, the frame similarity $FS(f_{t1}, f_{t2}, \delta)$ is adopted, varying $\delta$ value according to the set \{0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 0.9\}. The ideal method must have CUSa (or COV) equals to 1, and CUSe equals to zero. As can be seen, HSummary outperforms the compared methods regardless of the visual feature used (see $F$-measure values). Moreover, CUSa and COV for $HSummary_G$ is a little bit better than $HSummary_{BN}$; and the CUSa for VSUMM1 is better than HSummary, however its CUSe is much bigger. Furthermore, it is worth to mention that COV value for all methods is smaller than the CUSa value for the same method. This behavior could be related to the elimination of average step for computing COV.

In Fig. 7, and Fig. 8, static video summaries computed by several methods are illustrated. For comparing those results to the user summaries, $\delta$ is set to 0.5. For each method, $F$-measure value is shown. Considering these results, HSummary outperforms the compared methods. In some cases, there are redundant frames, which results in higher CUSe, and, consequently, lower $F$-measure values (see Fig. 7(a) and (d)).

VI. CONCLUSION AND FURTHER WORKS

Video summarization is a simplification of video content for compacting the video information in which a small number of frames must be selected for saliently representing its content. To address video summarization, we proposed the use of a hierarchical graph-based clustering method, called HSummary, which allows to apply a similarity measure between clusters during graph partition, instead of considering only the similarity between isolated frames. As in [16], the hierarchical method for partitioning the frame similarity graph using observation scales produces a hierarchy of partitions in which the clusters can easily be inferred. Another important contribution of this work is the definition of a new evaluation measure that assesses the diversity of opinions of user summaries, called Covering. Experimental results provides quantitative and qualitative comparison between the new approach and other popular algorithms from the literature, showing that the new algorithm is robust and efficient. Concerning quality measures, HSummary outperforms the compared methods regardless of the visual feature used in terms of $F$-measure. In order to improve and better understand our results, further works involve multimodal analysis, inclusion of new features and automatic scale identification. Furthermore, a n-gram study may help us in avoiding error in the identification of keyframes.

VII. ACKNOWLEDGMENTS

The authors are grateful to PUC Minas (Pontificia Universidade Catlica de Minas Gerais), MIC BH (Microsoft Innovation Center Belo Horizonte), CNPq, Capes and FAPEMIG for the financial support of this work.

REFERENCES

Figure 8. Examples of static video summary for the video 53 using several methods. For each result, the $F$-measure is shown in parentheses.


